# An ensemble-based smoother with retrospectively updated weights for highly nonlinear systems

T. M. Chin, M. J. Turmon, J. B. Jewell,
Jet Propulsion Laboratory, California Institute of Technology.
M. Ghil[1]
Department of Atmospheric and Oceanic Sciences and Institute of Geophysics and Planetary Physics, University of California, Los Angeles

March 3, 2006

Corresponding author: T.M. Chin
M/S 238-600, 4800 Oak Grove Drive, Pasadena, CA 91109, USA.
mike.chin@jpl.nasa.gov

.

[1]Additional affiliation: Département Terre-Atmosphère-Océan and Laboratoire de Météorologie Dynamique du CNRS, Ecole Normale Supérieure, F-75231 Paris Cedex 05, FRANCE

**Abstract**

Monte Carlo computational methods have been introduced into data assimilation for nonlinear systems in order to alleviate the computational burden of propagating the system's full posterior probability distribution. By propagating an ensemble of representative states, algorithms like the Ensemble Kalman Filter (EnKF) and Resampled Particle Filter (RPF) rely on the existing modeling infrastructure to approximate the distribution based on the evolution of this ensemble.

This work presents an ensemble-based smoother that is applicable to Monte Carlo filtering schemes like the EnKF and RPF. At the minor cost of retrospectively updating a set of weights for ensemble members, this smoother provides superior state tracking for two simple nonlinear problems, the double-well potential and the trivariate Lorenz system. The algorithm does not require retrospective adaptation of the ensemble members themselves, and it is thus suited to a streaming operational mode. The accuracy of the proposed backward-update scheme in estimating non-Gaussian distributions is evaluated by comparison to the more accurate estimates provided by a Markov chain Monte Carlo algorithm.

## 1. Introduction

Monte Carlo computational methods have been receiving a growing interest for application to sequential data assimilation (e.g., Evensen 1994; Houtekamer and Mitchell 1998; Keppenne and Rienecker 2002; Ott *et al.* 2004) because they are computationally more tractable than the conventional Kalman filter (Ghil 1997). Moreover, Monte Carlo algorithms can be implemented without the labor-intensive development, tuning, and validation of the *tangent linear model* and its adjoint (Errico 1997), which are major parts of the variational schemes. Furthermore, background fields can be chosen from a richer class that better represents real systems, including anisotropic covariances and non-Gaussian noise probability distributions. In contrast, standard optimal interpolation methods assume an isotropic covariance (Daley 1991), while the extended Kalman filter (eKF) assumes implicitly that the underlying probability distribution is Gaussian (Jazwinski 1970). Finally, Monte Carlo methods can be designed to deal correctly with nonlinear system dynamics that produce non-Gaussian posterior distributions.

Monte Carlo data assimilation schemes have been used mostly to obtain a *filtered analysis*, in which only data prior to the time of analysis are used. Examples are (i) the *Ensemble Kalman filter* (EnKF; Evensen 1994), which is a Monte Carlo approximation of the eKF; (ii) a non-Gaussian extension of the EnKF that uses a "Gaussian mixture" (weighted sum of Gaussian functions) instead of a single Gaussian probability density function to represent the background distribution (Anderson and Anderson 1999; Bengtsson *et al.* 2003; Kim *et al.* 2003); and (iii) the *particle filter* (Pham 2001; van Leeuwen 2003; Kim *et al.* 2003) that, unlike the previous two, is non-parametric. We will inclusively refer to these Monte Carlo filtering schemes as *ensemble filters* for simplicity.

In contrast to filtering, a *smoothed analysis* results from retrospective processing of the observed data, where all data available, both past and future, are incorporated into the analysis. Use of the future data increases the effective amount of data available for each analysis and can significantly reduce analysis errors (Cohn *et al.* 1994). Filtered analyses are most relevant for the initialization of numerical weather forecasts, where future data are not available operationally. Smoothed analyses including reanalysis products, on the other hand, have been providing some of the best available records for the evolution of the atmosphere and oceans (Bennett 1992; Wunsch 1996; Kalnay 2003) and allowing the scientific community to validate climate models and to conduct a variety of process studies. In this paper, we examine a Monte Carlo approach to smoothing; we note in passing that deterministic approaches exist (Eyink and Restrepo 2000). The method presented here allows one to convert the output of any of the ensemble filters mentioned above into a sequence of smoothed analyses, at a modest additional computational cost.

It is well known that an efficient forward–backward algorithm suffices to compute the

best *linear* state-space estimate (Jazwinski 1970; Gelb 1974). First, the optimal filtered state estimates are computed forward in time with the Kalman filter, and then these estimates are updated backward in time to find smoothed estimates. In this paper, we study a smoothing algorithm with a similar forward–backward structure, in which the forward phase can be any of the ensemble filters mentioned above. The backward phase of the algorithm uses the ensemble of the filtered trajectories as its input and sequentially associates a time-dependent probabilistic weight with each ensemble member. The probabilities are assigned based on the relative likelihood of local step-to-step variations, while the likelihood of each state transition is determined by the dynamic model and the state and observation noise distributions. The smoothing algorithm we present leaves the filtered ensemble members untouched — it only adjusts the weight of each ensemble member.

Monte Carlo smoothing schemes for data assimilation have been proposed in the past. For example, the smoother presented by van Leeuwen and Evensen (1996) is a weighted average of sample state trajectories, where the weights are determined by closeness of the corresponding trajectories to the data over the entire analysis interval. Since the weights are fixed for the entire interval, there is seldom enough model-permitted pliability in the resulting state trajectories to allow likely encounters with all of the data, especially when the interval is long and when the forecast equations contain significant errors and uncertainty. Evensen and van Leeuwen (2000) later improved this approach essentially by dividing the interval and performing data updates to the ensemble of trajectories sequentially, from one subinterval to the next. Each trajectory is updated using a large space-time covariance function that covers the subinterval from its beginning to the most recent data-update epoch.

In contrast, the smoothing algorithm presented here and initially explored by Kitagawa (1996), Hürzeler and Künsch (1998), and Doucet *et al.* (2000) relies on a formula that is sequential backward in time and is consistent with sampling of the ideal posterior distribution. While there are variants in execution of this algorithm (e.g., Godsill *et al.* 2004), the basic smoothing formula is widely applicable to the result from any of the ensemble filters mentioned previously. Qualitatively, the smoother examines each possible one-step state transition in the filtered ensemble and determines the likelihood of a transition taking place, given a stochastic dynamic model. The weight, or probability mass, assigned to each filtered state value is then adjusted sequentially according to this likelihood. In this paper, we demonstrate efficacy of the smoother in tracking shifts between dynamic regimes within two systems studied extensively in data assimilation. We also check the posterior means, variances, and marginal distributions computed by the smoother against ground truth supplied by an independently implemented Markov chain Monte Carlo (MCMC) method. By using convergence diagnostics that are easy to verify for the two example problems considered here, we ensure that the MCMC method provides accurate sampling of the posterior distributions.

We will show that such ground truth is represented more accurately by smoothed ensembles than by the corresponding filtered ensembles. The updates made by the smoother to filtered ensembles can thus lead to dramatic improvements in the ensemble-mean estimate of the state trajectory.

The paper is organized as follows. Section 2 summarizes the variety of approaches to Monte Carlo solutions to the filtering problem. Section 3 presents the backward-time phase of the smoother applicable to these filtered estimates. Section 4 outlines the MCMC method used to obtain the posterior reference distribution. The estimation algorithms are compared numerically using the two test problems described in section 5. The results of this comparison are described in section 6, followed by concluding remarks in section 7. Numerical details of the algorithms we use are given in three appendices.

## 2. Ensemble filters for nonlinear systems

We outline in this section the statistical approach to sequential data analysis and define some notations. Let the vector $\mathbf{x}_t$ be the state of the model at time index $t = 1, 2, \ldots, T$, and $\mathbf{y}_t$ be the corresponding instantaneous observation, if it is available. These variables evolve according to the fundamental rules

$$\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_{t-1}) + \mathbf{w}_t \tag{1}$$

$$\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t) + \mathbf{v}_t \tag{2}$$

given an initial state $\mathbf{x}_0$. In (1), $\mathbf{f}_t$ is the tendency function of the model and $\mathbf{w}_t$ is a zero-mean perturbation vector with a given probability distribution. This vector includes model inaccuracies, as well as stochastic effects on the system. In (2), $\mathbf{h}_t$ is the observation operator, and $\mathbf{v}_t$ is a zero-mean vector allowing for both sensor and sampling noise.

The observer accumulates data $\mathbf{Y}_t \equiv \{\mathbf{y}_{t'} : 1 \leq t' \leq t\}$ up to time $t$. The filtering and smoothing problems are solved by finding the posterior probability density function (PDF) conditioned on a particular observation record, $p(\mathbf{x}_t|\mathbf{Y}_{t'})$. The filtered analysis estimates the conditional PDF $p(\mathbf{x}_t|\mathbf{Y}_t)$, while the smoothed analysis estimates $p(\mathbf{x}_t|\mathbf{Y}_T)$. The desired summary analysis can then be obtained as a moment (e.g., the mean) or the mode (e.g., maximum posterior likelihood) of the corresponding posterior PDF. In this section we consider state representations and sequential Monte Carlo updates in the filtering problem, and leave smoothing for the next section.

### 2.1. Dynamic update in ensemble filters

Ensemble filter schemes represent the conditional PDF $p(\mathbf{x}_t|\mathbf{Y}_t)$ by a collection of $N$ samples, $\mathbf{x}_t^{(n)}$, $1 \leq n \leq N$. Each sample is updated sequentially in time. The essence of ensemble

filter schemes is their use of repeated, randomly perturbed forecasts to sample the evolution of the state equation under statistical uncertainty:

$$\mathbf{x}_t^{(n)} = \mathbf{f}_t(\mathbf{x}_{t-1}^{(n)}) + \mathbf{w}_t^{(n)} \qquad \text{for } n = 1, \ldots, N, \tag{3}$$

where $\mathbf{w}_t^{(n)}$ are independent samples of the perturbation vector $\mathbf{w}_t$, and each sample trajectory starts with a suitably selected $\mathbf{x}_0^{(n)}$. As mentioned previously, a major practical advantage of ensemble schemes is that the existing model code can be used to compute the tendency $\mathbf{f}_t$. When $\mathbf{w}_t \equiv 0$, eq.(3) reduces to a deterministic model prediction step.

### 2.2. Data update in ensemble filters

Various tradeoffs of computational cost versus modeling generality may be preferred depending on the degree of nonlinearity in (3). While all ensemble filters considered here use (3) to dynamically update the ensemble members, each filter uses a distinct technique to update the ensemble with the data, due to differing representations of $p(\mathbf{x}_t|\mathbf{Y}_t)$. These representation techniques are briefly summarized below. The actual numerical algorithms used in our experiments are detailed in appendix A.

**2.2.1. Gaussian parameterization** The EnKF (appendix A.1) assumes a Gaussian posterior and thus needs only the first two moments, the mean and covariance of the samples from (3), to approximate $p(\mathbf{x}_t|\mathbf{Y}_t)$. The Gaussian assumption is consistent with the underlying premise of the eKF algorithm that has motivated the EnKF development (Evensen 1994). Unlike the eKF, however, the EnKF does not rely on linearized dynamics to forecast the covariances.

**2.2.2. Gaussian mixture parameterization** Nonlinear dynamics $\mathbf{f}_t$ does not preserve a Gaussian initial PDF, thus making the Gaussian representation insufficient in principle. In consequence, as we will see, methods like the EnKF that use a Gaussian PDF representation may experience a saturation of performance despite unbounded increases in the ensemble size. A practical non-Gaussian parameterization of the conditional PDF is a sum of Gaussian basis functions, called a *Gaussian mixture* distribution (McLachlan and Peel 2000). Anderson and Anderson (1999), Bengtsson *et al.* (2003), and Kim *et al.* (2003) have experimented with Gaussian mixture parameterizations and have verified that filter performance can exceed that of a single Gaussian. The data-update formulas in this case are considerably more complex than the standard minimum-variance (Gauss-Markov) estimator used under Gaussian assumptions.

**2.2.3. Nonparametric approximation** Instead of assuming a functional form, nonparametric approaches to data updating represent the PDF using a collection of samples drawn from it. Each sample consists of a state value $\mathbf{x}_t^{(n)}$ and associated probability mass $p_t^{(n)}$, effectively approximating the desired conditional PDF by the probability mass function

$$p(\mathbf{x}_t \,|\, \mathbf{Y}_t) \approx \sum_{n=1}^{N} p_t^{(n)}\, \delta(\mathbf{x}_t - \mathbf{x}_t^{(n)}) \tag{4}$$

where $\delta$ is the Dirac delta function. Under this density, for example, the posterior mean is estimated by $\sum_{n=1}^{N} p_t^{(n)} \mathbf{x}_t^{(n)}$. The probability mass $p_t^{(n)}$ is computed from the posterior PDF via Bayes' formula and depends therefore on the data set $\mathbf{Y}_t$. This approach is known as the *particle filter* (appendix A.3) in the statistical literature (e.g., Kitagawa 1996) and has been experimented with in the context of data assimilation (Pham 2001; van Leeuwen 2003; Kim *et al.* 2003). Ensemble members inconsistent with the data receive low weight, eventually reducing the effective ensemble size. Periodic resampling of the ensemble is thus necessary to keep a diverse set of ensemble members (appendix A.4). The resulting method is referred to as the *resampled particle filter* (RPF).

## 3. Backward smoothing of particle weights

The smoothed analysis is based upon the posterior PDF $p(\mathbf{x}_t|\mathbf{Y}_T)$ for $T \geq t$, that is, the observation sets are analyzed retrospectively. In particular, a *fixed-interval* smoother computes $p(\mathbf{x}_t|\mathbf{Y}_T)$ for a fixed observation interval $T$, while the *fixed-lag* smoother finds $p(\mathbf{x}_t|\mathbf{Y}_{t+t'})$ for a fixed $t' > 0$ (Cohn *et al.* 1994). Because the algorithm we present is sequential in backward time, it applies equally well to both formulations, and we shall focus on the fixed-interval case hereafter.

Let $p_{t|t}(\mathbf{x}_t) \equiv p(\mathbf{x}_t \,|\, \mathbf{Y}_t)$ and $p_{t|T}(\mathbf{x}_t) \equiv p(\mathbf{x}_t \,|\, \mathbf{Y}_T)$ denote the filtered and smoothed conditional probability densities, respectively. It is well known that the smoothed density function $p_{t|T}$ can be evaluated by updating the filtered density $p_{t|t}$ using a backward sequential computation, under the assumption that the dynamical and observation equations are linear and the stochastic inputs to the model have Gaussian distributions (e.g., Jazwinski 1970). For the general case without the linear and Gaussian assumptions, the smoothed and filtered densities can be proportionally related (appendix B) as

$$p_{t|T}(\mathbf{x}_t) \propto p_{t|t}(\mathbf{x}_t)\, p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \,|\, \mathbf{x}_t), \tag{5}$$

where the last factor is the probability that the present state $\mathbf{x}_t$ would give rise to the (known) future observation sequence. This probability can be expressed in terms of the the future smoothed density $p_{t+1|T}$ by applications of the chain rule and Bayes rule. In particular, the

following recursion formula can be derived (see appendix B):

$$p_{t|T}(\mathbf{x}_t) = \int p_{t+1|T}(\mathbf{x}_{t+1}) \frac{p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t)\, p_{t|t}(\mathbf{x}_t)}{\int p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t)\, p_{t|t}(\mathbf{x}_t)\, d\mathbf{x}_t}\, d\mathbf{x}_{t+1} \quad . \tag{6}$$

This formula can be discretized, using (4), as

$$p^{(n)}_{t|T} = \sum_{m=1}^{N} p^{(m)}_{t+1|T} \frac{p(\mathbf{x}^{(m)}_{t+1} \,|\, \mathbf{x}^{(n)}_t)\, p^{(n)}_{t|t}}{\sum_{\ell=1}^{N} p(\mathbf{x}^{(m)}_{t+1} \,|\, \mathbf{x}^{(\ell)}_t)\, p^{(\ell)}_{t|t}} \quad , \tag{7}$$

and allows the evaluation of $p_{t|T}$ given $p_{t+1|T}$, and hence a backward sequential computation of the smoothed densities. The resulting computational scheme is referred to as the *backward sequential smoother* (BSS).

The BSS algorithm updates the weights in the weighted ensemble of filtered trajectories $\left\{(\mathbf{x}^{(n)}_t, p^{(n)}_{t|t}) \colon 1 \le n \le N\right\}$; it thus refines ensemble filtering algorithms like the EnKF and RPF, in which the ensemble members are equally weighted: $p^{(n)}_{t|t} = 1/N$. By definition, the filtered and smoothed ensembles are identical at $t = T$. The smoothed weights $p^{(n)}_{t|T}$ are then computed backward in time, by applying (7) for $t = T-1, T-2, \ldots, 1$. Application of the BSS scheme is straightforward when the state-transition probabilities $p(\mathbf{x}^{(m)}_{t+1} \,|\, \mathbf{x}^{(n)}_t)$ for $1 \le m, n \le N$ can be readily evaluated. For the additive dynamic perturbation of (3), the transition from $\mathbf{x}^{(n)}_t$ to $\mathbf{x}^{(m)}_{t+1}$ implies $\mathbf{x}^{(m)}_{t+1} = \mathbf{f}_t(\mathbf{x}^{(n)}_t) + \mathbf{w}_{t+1}$, so the probability is obtained by evaluating the density of $\mathbf{w}_{t+1}$ at $\mathbf{x}^{(m)}_{t+1} - \mathbf{f}_t(\mathbf{x}^{(n)}_t)$. The BSS algorithm thus uses the state-transition rule to impose dynamical constraints on the filtered weights $p^{(n)}_{t|t}$; in particular, it reduces the weights of those trajectories that contain jumps at the epochs of data updates, which are inconsistent with the model dynamics.

The BSS algorithm requires three new tasks: retention of the filtered ensemble trajectories, evaluation of the state transition probabilities, and actual computation of the smoothed probability weights. Regarding the first, which is a storage requirement, we note that the filtered trajectories $\mathbf{x}^{(n)}_t$ themselves are unaltered by the backward smoothing pass. In fact, these ensemble members are examined serially, in a streaming mode, once forward in time and once backward. The ensemble trajectories do not ever have to be simultaneously present in primary memory. The storage requirement of the algorithm (in terms of its computational cache) is hence $O(ND)$, where $D$ is the dimension of the model state vector.

The second requirement, computing the $N \times N$ state transition matrix $p(\mathbf{x}^{(m)}_{t+1} \,|\, \mathbf{x}^{(n)}_t)$, requires $N^2$ density evaluations for the typical case of additive state noise. Computation of the matrix is facilitated by storing the one-step-ahead forecasts $\mathbf{f}_t(\mathbf{x}^{(n)}_t)$, which doubles the storage requirement.

The third requirement, recursive evaluation of the smoothed weights, is fundamentally computational. Specifically, eq.(7) amounts to two matrix-vector multiplications. The total

computational cost of the algorithm is $O(TN^2)$, which is insignificant for typical ensemble sizes $N < 10^3$. If its share became significant relative to model computations, sparsity in the transition matrix could be exploited. The algorithm thus requires minimal computational resources over those allocated to the ensemble filter.

## 4. Markov chain Monte Carlo (MCMC) estimation

The BSS procedure above estimates the posterior distribution $p(\mathbf{x}_t \,|\, \mathbf{Y}_T)$. As an independent reference to evaluate the ensemble smoothers, we also use an MCMC estimate of the posterior PDF. Our MCMC computations do not directly assess accuracy of the filters, which estimate $p(\mathbf{x}_t \,|\, \mathbf{Y}_t)$, but they do assess the information loss due to the filter's use of $\mathbf{Y}_t$ rather than $\mathbf{Y}_T$.

MCMC is an old technique for sampling from complex probability distributions (Metropolis *et al.* 1953) that has received much recent attention (Gilks *et al.* 1996; Robert and Casella 2005) as computational resources and enhancements to the underlying principles have put it within reach of many applications. It is well-known that the simple, well-understood Metropolis-Hastings algorithm used for MCMC in this paper does not scale to realistic data assimilation problems. However, versions of MCMC have been used in large-scale spatial inversion problems (Turmon *et al.* 2002; Haario *et al.* 2004), and there is potential to use MCMC variants that exploit the sequential structure of the data assimilation problem in the more challenging space-time setting (Alexander *et al.* 2005). As we shall see, even simple variants of MCMC are quite effective for the small systems considered here, and we need no special twists to generate accurate estimates of the posterior.

Ensemble filters and smoothers, as well as MCMC, all rely on random sampling, but the latter works very differently from ensemble-based methods. MCMC fits a candidate state sequence by taking many passes through the time series, so it is able to adapt the trajectory with reference to all observed data, avoiding the loss-of-diversity problem experienced by the particle filter. This property makes MCMC a good check of the BSS posterior estimate. Furthermore, additional MCMC samples can be generated simply by running the sampler longer. In this study, this has allowed us to generate very large MCMC sample sets to ensure the posterior distribution is sampled thoroughly.

Other methods for obtaining the exact PDF evolution have been used in the literature. The posterior distribution is defined by the Fokker-Planck equation and Bayes' rule (Jazwinski 1970). Computing it directly through these equations, however, requires multidimensional integrals over the state variables $\mathbf{x}_t$. In one dimension, for example, these integral equations have been solved directly using numerical techniques (Miller *et al.* 1999). The advantages of MCMC relative to such a direct solution are in its ease of implementation in the multidi-

mensional setting, well-characterized performance properties, and a theoretical guarantee of convergence to the true posterior PDF.

To see how MCMC works, consider estimation of a statistical moment $\phi(\mathbf{z})$, such as the mean or covariance, of the random vector sequence $\mathbf{z} = (\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_T)$ which in our case is the state sequence that represents the entire trajectory. The moment can be estimated via a *sample average* from the distribution $\pi(\mathbf{z}) \equiv p(\mathbf{z} \,|\, \mathbf{Y}_T)$.

$$E\,\phi(\mathbf{z}) = \int \phi(\mathbf{z})\,\pi(\mathbf{z})\,d\mathbf{z} \simeq \frac{1}{N}\sum_{n=1}^{N} \phi(\mathbf{z}^{(n)}) \tag{8}$$

where $E$ is the expectation operator and each sample trajectory $\mathbf{z}^{(n)}$ is drawn from the PDF $\pi$. Ideally, the $N$ samples are drawn independently, but in many situations this is not easy. In MCMC, simple procedures are followed to define a Markov chain that has $\pi$ as its stationary distribution. The chain is started at $\mathbf{z}^{(0)}$, and after an initial spin-up period, each sample from the chain is approximately a draw from $\pi$. Even though the samples are not independent, their sample average (8) still converges to the the correct mean; that is, the chain is constructed to be ergodic. It can be shown (Tierney 1996) that, as $N \to \infty$, the Markov chain converges to the correct posterior PDF, independently from the initial trajectory estimate $\mathbf{z}^{(0)}$. Diagnostics based on inter-sample correlation can help determine when convergence has occurred.

In this paper, we construct the Markov chain according to the Metropolis-Hastings method (see Hastings 1970, and appendix C here). We start with an essentially arbitrary trajectory $\mathbf{z}^{(0)}$, and perform thousands of updates of $\mathbf{z}$ to stabilize the posterior PDF. One update to $\mathbf{z}$ requires sweeping over its $T$ entries, in order to update each component $\mathbf{x}_t$ according to a randomized Metropolis-Hastings rule. After this spin-up period, further sweeps through the full data set are performed to gather $N$ samples of $\mathbf{z}$ and compute statistics of the posterior density via (8). To compute the posterior mean, we set $\phi(\mathbf{z}) = \mathbf{z}$. Other statistics, such as variances and histograms, are computed just as easily from the same $N$ samples.

## 5. Evaluating algorithm performance

Many performance characteristics of filtering and smoothing algorithms can be illustrated in their application to two simple nonlinear systems: the double-well potential and the trivariate Lorenz (1963) model. Both exhibit phase transitions and have been used extensively as reference problems in the nonlinear assimilation literature (Miller *et al.* 1994; Miller *et al.* 1999; Anderson and Anderson 1999; Evensen and van Leeuwen 2000; Pham 2001; Kim *et al.* 2003).

The double-well system is one of the simplest nonlinear dynamical systems that can be used to study tracking of phase transitions. The motivating idea is a particle in a potential double-well $F(x) = x^4 - 2x^2$, with minima at $x = \pm 1$, subject to stochastic forcing so that transitions between one well and the other take place from time to time. Formally, this behavior is modeled by the continuous-time diffusion process $dx_t = g(x)\,dt + \kappa\,dB_t$, where $g(x) = -dF(x)/dx$, $B_t$ is the unit-variance Brownian motion process, and $\kappa$ determines the strength of the *dynamic noise*. Following Miller *et al.* (1994), we use $\kappa = 0.5$. The stationary distribution of the diffusion process is proportional to $\exp(-2F(x)/\kappa^2)$, placing equal mass at the two minima.

The trivariate Lorenz model, abbreviated hereafter as L63, has also been extensively studied for assimilation purposes and may be written as $d\mathbf{x}/dt = \mathbf{g}(\mathbf{x})$, where

$$\mathbf{x} \equiv \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} 10(x_2 - x_1) \\ 28x_1 - x_2 - x_1x_3 \\ x_1x_2 - (8/3)x_3 \end{bmatrix}.$$

The system's attractor is known to have a fractal dimension slightly higher than two, and its phase-space geometry (the two "butterfly wings") is composed of two dynamic regimes in a qualitative sense.

For numerical experiments, the system equations (1) and (2) are constructed as follows: We discretize both the double-well and L63 diffusion dynamics at a fixed time interval $\tau$ as $\mathbf{x}_t = \mathbf{x}_{t-1} + \tau\mathbf{g}(\mathbf{x}_{t-1}) + \mathbf{w}_t$, where $\mathbf{w}_t$ is Gaussian distributed with zero mean and covariance matrix $\kappa^2\tau\mathsf{I}$, where $\mathsf{I}$ is the identity matrix. For L63, we have $\kappa = 0$, but a non-zero $\kappa$ may be used to simulate imprecise physics. The observation equation is $\mathbf{y}_t = \mathbf{x}_t + \mathbf{v}_t$, where $\mathbf{y}_t$ is the noisy observation which may not be available at all times, and $\mathbf{v}_t$ is Gaussian-distributed, zero-mean, and with a covariance matrix of $\sigma^2\mathsf{I}$.

Our experiments are directed at understanding the benefits of adopting a non-Gaussian PDF representation, as well as evaluating the reduction in estimation uncertainty obtained by the BSS smoothing approach outlined in section 3. Two filtering algorithms, EnKF and RPF, are compared against their respective smoothing counterparts. We use the MCMC smoother, run to stationarity, as a check on the posterior distributions computed by these algorithms. In particular, time series of the mean and standard deviation from filtered and smoothed ensembles are compared to those from MCMC sample averages. Also, snapshots of the state distribution, approximated by normalized histograms of ensemble values at a fixed time, from a filter and smoother are compared to the corresponding distributions computed by MCMC.

We have also compared snapshot distributions of filtered and smoothed ensembles using the order-statistic histogram, often called a Talagrand diagram, to determine if the true state and the ensemble members are statistically indistinguishable at a given time (Anderson

and Anderson 1999; Lawson and Hansen 2004). A single order statistic is computed by finding the rank of the true state value $\mathbf{x}_t$ within the sorted ensemble of $N$ state values $\{\mathbf{x}_t^{(n)} : 1 \le n \le N\}$ at the given time. Then, by repeating the filter or smoother analysis $10^3$ times, we construct a histogram of these order statistics. A necessary condition for the true posterior and ensemble distribution to be identical is that the order statistic be uniformly distributed between 0 and $N$. We used the second component of the $\mathbf{x}_t$ variable in the L63 system for this test. Finally, the accuracies of the filtered and smoothed estimates are evaluated using the standard root-mean-square (RMS) criterion, applied to the difference between the ensemble mean and true trajectories.

## 6. Numerical results

### 6.1. Double-well results

We obtained a sample trajectory with noisy observations by simulating the double-well system with $\kappa = 0.5$ and $\sigma = 0.2$, implemented with $\tau = 0.05$. In the sample, the state variable makes a descending transition, from the well centered at $+1$ to the other at $-1$, followed by an ascending transition back into the original well. Each visit is observed sparsely by three or four data points (Fig. 1a). Given these 11 observations, the trajectory is reconstructed using the ensemble filter, BSS smoother, and MCMC smoother. Here, the RPF algorithm ($N = 10^4$ particles) is used to obtain the filter results, to which BSS is applied to obtain the smoother results. The MCMC posterior distribution is obtained by initializing ("spinning-up") the Metropolis-Hastings Markov chain for $10^5$ sweeps and then storing 2000 sample trajectories, skipping 2000 sweeps between adjacent stored samples. The large ensemble size ($N = 10^4$) and MCMC sweep number ($4.1 \times 10^6$) are used to compute well-sampled histograms of the posterior distribution (see below), but are not necessary just to track the state transitions accurately.

Fig. 1

The filtered (RPF) and smoothed (RPF-BSS) ensemble means, or $\sum_{n=1}^{N} p_{t|t}^{(n)} \mathbf{x}_t^{(n)}$ and $\sum_{n=1}^{N} p_{t|T}^{(n)} \mathbf{x}_t^{(n)}$ respectively, as well as the mean MCMC trajectory have successfully reconstructed the state history (Fig. 1b). The RPF mean shifts abruptly at the first observation after each transition, as expected from the sequential nature of the algorithm. The RPF-BSS and MCMC means are qualitatively similar and are better approximations to the actual state trajectory, due to availability of the retrospective observations. Note also that the RPF-BSS and MCMC means use the entire time between observations to make the phase transition, as dictated by the double-well dynamics. This choice spreads the necessary large values of state noise, which is penalized quadratically by the Gaussian distribution of $\mathbf{w}_t$, out into a larger number of smaller values for a lower cumulative cost in energy (or log-probability).

The estimated standard deviations of all three methods (Fig. 1c) show a similar level

of background variability (slightly less than 0.2), representing the jitter in the system while visiting a well. The variance found by the MCMC analysis increases markedly toward unity during the two phase transitions. The increased uncertainty, extending across the entire span between observations, is due to the weak constraint supplied by the dynamics in-between the two potential wells. The RPF-BSS variance has the same features and agrees closely with the very accurate MCMC estimate. On the other hand, the RPF variance completely misses the significance of the phase transitions with respect to the variance of the state estimate. Fig. 2

For a strongly non-Gaussian posterior distribution, the first two moments do not tell the whole story. We see from the MCMC posterior distribution near phase transition (Fig. 2) that the probability mass is actually spread out widely with concentrations around the stable points $x = \pm 1$. The posterior distribution is clearly non-Gaussian at these times. The normalized histogram of the RPF ensemble at any given time is, however, clustered tightly near the estimated mean, moving almost all the particles toward the latest observed value. The BSS smoother, on the other hand, re-evaluates the weights of the filtered particles so that the resulting distribution becomes more widely spread and shows clear signs of trying to approximate a bimodal distribution. Recall that the BSS does not alter the ensemble members: it simply assigns a new weight to each existing member at each time. So, the BSS density approximation is smoothest near the current mode ($x = +1$ in Fig. 2a versus $x = -1$ in Fig. 2b). Around the other mode, only a few ensemble members must carry all the probability mass and the approximation is coarser. However, note that the relative posterior probability carried by the two modes is roughly correct. Away from the phase-transition times, the posterior distributions for the three methods agree closely, and are nearly Gaussian, centered around the estimated means (not shown).

Because the RPF and BSS algorithms allow mass to be placed at or near the correct locations in state space, we would expect their tracking of state changes to be superior to a method based on Gaussian PDF reconstructions alone. The particle methods, however, depend on statistically unlikely members to maintain the non-Gaussian features of their PDFs, and these features become apparent only during phase transitions. Maintenance of such minority members by chance is more difficult when the ensemble size is limited, as is the case in current data assimilation practice. The tracking abilities of an ensemble filter and smoother can both deteriorate significantly when the ensemble size is reduced to $N = 100$ (Figs. 3a,b). Fig. 3

Practical methods to remedy this computational limitation exist. One approach is to view any realization of the system noise as a brute-force search for the optimal state trajectory. Thus, when the ensemble size is limited, one can compensate by expanding the search interval. A simple procedure is thus to inflate the system noise variance beyond its known value, and this has indeed proven effective. Specifically, when the noise-amplitude

parameter $\kappa$ is increased by 40%, the mean trajectory, as well as the uncertainty estimate, can become quite adequate (Figs. 3c,d), especially for the ensemble smoother. In practice, this parameter of the algorithm, which determines the effective noise it uses, needs to be optimized. As illustrated in Figure 4, a strong enough noise allows the ensemble trajectories to capture the state transition. Excessive inflation of noise, however, degrades the analysis accuracy, especially at steady-state, by introducing too many dynamically unlikely ensemble members. The value of the dynamic noise variance thus serves as a regularization parameter for the purpose of trajectory estimation.

Fig. 4

### 6.2. Lorenz model results

Following the reasoning associated with $\kappa$ and Figure 4, we have experimented with L63, which has a well-documented sensitivity to the initial state, by simulating the system deterministically (i.e., $\kappa = 0$) to obtain a sample state-trajectory and noisy observations to which the ensemble filter and smoother are then applied by using a stochastic dynamics ($\kappa^2 = 0.1$). Randomly generated initial states are used for both the filter and smoother. The observations are provided only for two of the three state variables ($x_1$ and $x_3$) at an interval of 0.5 time unit, which is roughly the nominal Nyquist interval for the fastest periodic component, and the observation noise parameter was $\sigma^2 = 2$.

The RPF and EnKF algorithms are used for filtering, each with an ensemble size of $N = 40$; BSS is then used to smooth each of these two filtered trajectory ensembles. Figures 5–7 show that even with such a small ensemble size the smoother can track the state trajectory quite well. While the filters (two upper panels) also display some level of tracking capabilities, it is remarkable that the smoothers (two lower panels) have accurately reproduced some of the singular features that are not apparent from the sparse observations. In particular, the peaks at around times 27 and 39 occur between adjacent observations, and are missed entirely or underestimated by the filters, but are reconstructed accurately by the smoothers. This demonstrates that the BSS smoothing algorithm can efficiently determine the weights of the states in the filtered ensemble to improve the estimated ensemble mean. It is a striking illustration of how much information is latent in the nonlinear dynamics, unused by the filter, but recoverable by a correctly designed smoother.

Fig. 5
Fig. 6
Fig. 7

Comparison to MCMC confirms that the variance of the smoothed ensemble is smaller than the filtered variance and is closer to the MCMC variance (Fig. 8, top panel). Also, the posterior distribution of the $x_2$ variable is examined at the times indicated by arrows in Figure 6 for the RPF, RPF-BSS, and MCMC method (Fig. 8, bottom four panels), where the distribution is estimated as a normalized histogram of samples in each ensemble. These estimated distributions reveal that the smoothed ensemble in the L63 case is a close approximation of the correct distribution obtained by the MCMC technique.

Fig. 8

The Talagrand diagrams of the $x_2$ variable are also evaluated at the times indicated by arrows in Figure 6 and plotted in Figure 9. As a necessary condition for the ensemble to be unbiased, the order statistics must have a uniform distribution. Indeed, the smoothed analysis yields flatter plots than the filtered analysis. The unevenness of the histogram indicates that the ensemble either under- or over-estimates the truth as the distribution shifts to the left or right, respectively, in the plots. These plots show that, although the smoother inherits this problem from the filter, it does improve upon it. This observation is consistent with the fact that the smoother uses the identical ensemble as the filter, only with different weights assigned to each ensemble member. Fig. 9 Fig. 10

Figure 10 shows the RMS errors of the posterior mean produced by the various L63 analyses, as a function of ensemble size $N$. The RPF, EnKF, RPF-BSS and EnKF-BSS methods are compared. Clearly, the smoother always yields smaller RMS errors than the corresponding filter, due to the availability of retrospective data. When the ensemble size is small, the EnKF-based filter and smoother yield lower errors than the methods based on RPF. The reason for this is that the EnKF actively re-populates its ensemble around the best linear estimate at each update, while the RPF depends passively on random generation for members that happen to agree with data. Hence the RPF-based methods require a larger ensemble to randomly acquire members worthy of large weights. On the other hand, due to the underlying Gaussian assumption, the performances of EnKF and EnKF-BSS saturate after $N = 50$, showing virtually no improvements as $N$ is increased further. By contrast, it is known that RPF and BSS are asymptotically consistent, that is, as $N \to \infty$, the estimated posterior mean approaches the true Bayesian posterior mean at a rate proportional to $1/\sqrt{N}$ (Crisan 2001).

## 7. Concluding remarks

A backward sequential smoother (BSS) algorithm updates the weights assigned to each state sample generated by an ensemble filter. Significant improvement in accuracy over the ensemble-filter results has been demonstrated for both the Ensemble Kalman filter (EnKF) and Resampled Particle Filter (RPF). The BSS results are also tested against results from a Markov chain Monte Carlo (MCMC) algorithm.

In the case study using the Lorenz (1963, L63) model, we showed that the smoothing algorithm accurately reconstructs several state-trajectory features that are not apparent from the sparse observations and have been missed almost completely by the filtered mean trajectory (Figs. 5-7). Also, in the double-well case, the smoothed ensemble variance is seen to be consistently high during the state transitions where the uncertainty in the estimate is high, while the filtered variance does not present such information (Fig. 1). The increased

BSS variance at these times agrees quantitatively with the posterior variance as computed by the MCMC algorithm. Finally, highly non-Gaussian features are apparent in the posterior distributions at transition times, and the BSS tracks these successfully in both test cases (Figs. 2 and 8).

Our results from both the Gaussian (EnKF) and non-Gaussian (RPF) filters (and corresponding BSS smoothers) depend quite strongly on the ensemble size $N$ (Fig. 10). In the L63 case, the performance of the Gaussian filter or smoother saturates quickly as $N$ increases, while the performance of their non-Gaussian counterparts improves steadily with $N$. Evidently, the Gaussian approximation is ultimately inadequate for the nonlinear dynamics. The non-Gaussian RPF algorithm, however, requires a large $N$ to perform at its potential, as demonstrated in the double-well case as well (Fig. 3); this is the practical consequence of the slow, $1/\sqrt{N}$ convergence of the algorithm. It follows that, when the ensemble size $N$ is limited by computational resources, the Gaussian EnKF method and the associated BSS smoother may still provide acceptable performance in practice, even for a nonlinear model with non-Gaussian state distribution.

The RPF-BSS smoother reconstructs the true posterior PDF with an error proportional to $1/\sqrt{N}$. In practice, however, specific implementations may have different constants of proportionality, and ingenious sampling schemes may even lead to better convergence as a function of $N$. To this end, two important issues in ensemble filtering are: (i) sampling from the unknown initial PDF, and (ii) maintaining sufficient spread of the numerically evolving PDF to keep sampling the non-Gaussian features of the true PDF. The overall purpose is to generate perturbations that are effective in guiding the perturbed trajectories in the most likely directions of system evolution (Miller and Ehret 2002). Two approaches that are widely used in numerical weather prediction to achieve this purpose are bred vectors (Toth and Kalnay 1993; Kalnay 2003) and singular vectors (Molteni *et al.* 1996; Ehrendorfer and Tribbia 1997).

To the extent that the RPF-BSS smoother does not modify the trajectories of the RPF filter, the smoother simply inherits the quality of the sampling from the filter, although it does improve upon the weights attached to each trajectory at any given time. The smoother results in this paper highlight the fact that the perturbations used by the filter do not necessarily aim to mimic the state noise process $\mathbf{w}_t$; they represent, instead, a numerical device to improve the sampling efficiency of the finite set of particle trajectories in representing an elusive and evolving state-space distribution. We have thus shown that, when $N$ is fixed and small, both the filter and smoother performances can be improved significantly by enhancing the dynamic noise parameter value $\kappa$ (Fig. 4).

This observation motivates a future investigation on strategies for optimal perturbations and optimal sampling. The latter may involve variance reduction methods that are fairly well

known in the statistical literature but have been only partially explored in meteorological or oceanographic applications (Balgovind *et al.* 1983; Doucet *et al.* 2001). The loss of diversity in the ensemble can be measured during filtering. It is then possible to intervene and enhance the ensemble quality, for example, with MCMC-based tuning of members of the ensemble (Godsill and Clapp 2001). We expect to pursue some of these ideas in future work, in order to bring the smoothing algorithm presented here closer to implementation on realistic models of geophysical fluids.

## Acknowledgment

The research of TMC, JBJ and MJT was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (NASA). The research of MG at UCLA was supported by a grant from NASA's Physical Oceanography Program.

# APPENDIX

## A. Ensemble-based data-update schemes

The dynamic update step is given by (3) and is common to all filters. The data-update step is different in each filter scheme and is summarized here. In the observation equation (2), let $\mathbf{v}_t$ have a zero-mean Gaussian distribution with a given covariance matrix $\mathsf{R}_t$.

### A.1. The Ensemble Kalman filter (EnKF)

At each data update, the sample mean and covariances are computed according to

$$\overline{\mathbf{x}}_t = \frac{1}{N-1}\sum_{n=1}^{N}\mathbf{x}_t^{(n)} \ , \tag{9}$$

$$\overline{\mathsf{P}}_t = \frac{1}{N-1}\sum_{n=1}^{N}[\mathbf{x}_t^{(n)}-\overline{\mathbf{x}}_t][\mathbf{x}_t^{(n)}-\overline{\mathbf{x}}_t]^T \ . \tag{10}$$

In practice, when the ensemble size is small, some local approximation procedures may have to be applied to (10) to suppress spurious long-distance correlations (Houtekamer and Mitchell 1998; Ott *et al.* 2004). The mean vector and covariance matrix so obtained are

then simply substituted into the standard data-update step of the eKF algorithm:

$$\mathsf{K}_t = \overline{\mathsf{P}}_t \mathsf{H}_t^T \left( \mathsf{H}_t \overline{\mathsf{P}}_t \mathsf{H}_t^T + \mathsf{R}_t \right)^{-1} \tag{11}$$

$$\widehat{\mathbf{x}}_t = \overline{\mathbf{x}}_t + \mathsf{K}_t \left[ \mathbf{y}_t - \mathbf{h}_t(\overline{\mathbf{x}}_t) \right] \tag{12}$$

$$\widehat{\mathsf{P}}_t = \overline{\mathsf{P}}_t - \mathsf{K}_t \mathsf{H}_t \overline{\mathsf{P}}_t \tag{13}$$

where $\mathsf{H}_t \equiv \frac{\partial \mathbf{h}_t}{\partial \mathbf{x}}\big|_{\overline{\mathbf{x}}_t}$ is the Jacobian of the observation operator $\mathbf{h}_t$ about the background mean $\overline{\mathbf{x}}_t$. The updated mean $\widehat{\mathbf{x}}_t$ and covariance $\widehat{\mathsf{P}}_t$ are the filtered analysis.

A zero-mean Gaussian distribution with the resulting covariance $\widehat{\mathsf{P}}_t$ is sampled $N$ times to generate the perturbations $\Delta\widehat{\mathbf{x}}_t^{(n)}$ for $n = 1, \ldots, N$. The state samples are then updated as

$$\mathbf{x}_t^{(n)} = \widehat{\mathbf{x}}_t + \Delta\widehat{\mathbf{x}}_t^{(n)} \tag{14}$$

to complete the data-update step of EnKF.

### A.2. "Stochastic" EnKF

EnKF tends to underestimate the covariances of the analyzed state samples (14), when compared against the formal covariances computed from the full eKF equations (Burgers *et al.* 1998; Evensen 2003). A method to remedy this is to artificially corrupt the observed data $\mathbf{y}_t$ by samples $\mathbf{v}_t^{(n)}$ of the observation noise vector $\mathbf{v}_t$, and this "stochastic" technique is found particularly effective for nonlinear dynamics (Lawson and Hansen 2004). Specifically, the data-update steps (12)–(14) are replaced by

$$\mathbf{x}_t^{(n)} = \overline{\mathbf{x}}_t + \mathsf{K}_t \left[ \mathbf{y}_t - \mathbf{h}_t(\overline{\mathbf{x}}_t) + \mathbf{v}_t^{(n)} \right], \tag{15}$$

and the mean and covariance of the resulting state samples are computed as the filtered analysis and its covariance. We have found this technique effective for the EnKF estimation in the L63 case, when the ensemble size is small.

### A.3. Particle filter

In the particle filter, each state sample $\mathbf{x}_t^{(n)}$ is associated with a probability mass (weight) $p_t^{(n)}$, which is updated by persistence

$$p_{t|t-1}^{(n)} = p_{t-1}^{(n)} \tag{16}$$

along with the dynamic update formula (3).

The data-update step in the particle filter is simply to sample the Bayes formula. Specifically, the ensemble of sample–weight pairs $\left( \mathbf{x}_t^{(n)}, p_{t|t-1}^{(n)} \right)$ resulting from the prediction steps

(3), (16) is a discrete representation of the conditional probability function $p(\mathbf{x}_t|\mathbf{Y}_{t-1})$. To update the ensemble with the most recent data $\mathbf{y}_t$, we use the Bayes formula

$$p(\mathbf{x}_t|\mathbf{Y}_t) \quad = \quad p(\mathbf{y}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{Y}_{t-1}) \,/\, p(\mathbf{y}_t) \ , \tag{17}$$

where $p(\mathbf{y}_t)$ is constant with respect to $n$ and $p(\mathbf{y}_t|\mathbf{x}_t)$ can be easily evaluated using the distribution function for $\mathbf{v}_t$. Given (2) with Gaussian observation noise, we have

$$\begin{aligned} p(\mathbf{y}_t|\mathbf{x}_t) &= p[\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_t^{(n)})] \\ &\propto \exp\{-[\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_t^{(n)})]^T \mathsf{R}_t^{-1} [\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_t^{(n)})]/2\} \ . \end{aligned}$$

Thus, (17) becomes

$$p_t^{(n)} \quad = \quad c \cdot p_{t|t-1}^{(n)} \cdot \exp\{-[\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_t^{(n)})]^T \mathsf{R}_t^{-1} [\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_t^{(n)})]/2\} \ , \tag{18}$$

where $c$ is a normalization constant chosen so that $\sum_{n=1}^{N} p_t^{(n)} = 1$. In effect, the $N$ weights are overlaid with the likelihood of the data.

### A.4. Resampled particle filter (RPF)

The particle filter tends to become less efficient and eventually not effective in time due to the "empty-space phenomenon", where much of the probability mass concentrates in a very small number of state samples, ignoring most of the state space (Anderson and Anderson 1999). An effective remedy is to resample the updated particles $\mathbf{x}_t^{(n)}$, essentially pruning the highly unlikely state values, while retaining more likely values. The resampling procedure is: (i) update $\mathbf{x}_t^{(n)}$ via (3) and then $p_t^{(n)}$ via (18); (ii) obtain $N$ samples from the PDF (4), allowing for duplicate sample values; (iii) assign the $N$ samples as the new particles $\mathbf{x}_t^{(n)}$, and let the corresponding probability masses be $p_t^{(n)} = 1/N$ (a uniform distribution). The resulting algorithm is referred to as the *resampled particle filter*. The multinomial sampling of (ii) introduces extra variance in the Monte Carlo approximation process compared to other schemes that more systematically choose how often to include each sample. We did not observe performance losses due to this particular design choice, but differences have been observed in other applications (Liu and Chen 1998).

## B. Derivation of the Backward Sequential Smoother

We denote the consecutive observation set $\{\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T\}$ as $\mathbf{Y}_{t+1:T}$. When $u$ is either $t$ or $t+1$, the chain rule and Bayes rule yield

$$\begin{aligned} p(\mathbf{x}_u|\mathbf{Y}_T) &= p(\mathbf{x}_u|\mathbf{Y}_t, \mathbf{Y}_{t+1:T}) \\ &= p(\mathbf{x}_u, \mathbf{Y}_{t+1:T}|\mathbf{Y}_t)/p(\mathbf{Y}_{t+1:T}|\mathbf{Y}_t) \\ &= p(\mathbf{x}_u|\mathbf{Y}_t)\, p(\mathbf{Y}_{t+1:T}|\mathbf{x}_u, \mathbf{Y}_t)/p(\mathbf{Y}_{t+1:T}|\mathbf{Y}_t) \\ &= p(\mathbf{x}_u|\mathbf{Y}_t)\, p(\mathbf{y}_{t+1:T}|\mathbf{x}_u)/p(\mathbf{Y}_{t+1:T}|\mathbf{Y}_t) \ , \end{aligned} \tag{19}$$

where the last equality holds because the past and future observations are conditionally independent given the present state. For $u = t$,

$$p_{t|T} = p_{t|t} \ \ p(\mathbf{Y}_{t+1:T}|\mathbf{x}_t) \,/\, p(\mathbf{Y}_{t+1:T}|\mathbf{Y}_t) \ , \tag{20}$$

from which (5) follows. Also, by setting $u = t+1$ in (19), we have

$$p_{t+1|T} \,/\, p(\mathbf{x}_{t+1}|\mathbf{Y}_t) = p(\mathbf{Y}_{t+1:T}|\mathbf{x}_{t+1}) \,/\, p(\mathbf{Y}_{t+1:T}|\mathbf{Y}_t) \ . \tag{21}$$

The second term on the right-hand side of (20) can be written as

$$\begin{aligned} p(\mathbf{Y}_{t+1:T}|\mathbf{x}_t) &= \int p(\mathbf{Y}_{t+1:T}, \mathbf{x}_{t+1}|\mathbf{x}_t)\, d\mathbf{x}_{t+1} \\ &= \int p(\mathbf{Y}_{t+1:T}|\mathbf{x}_{t+1}, \mathbf{x}_t)\, p(\mathbf{x}_{t+1}|\mathbf{x}_t)\, d\mathbf{x}_{t+1} \\ &= \int p(\mathbf{Y}_{t+1:T}|\mathbf{x}_{t+1})\, p(\mathbf{x}_{t+1}|\mathbf{x}_t)\, d\mathbf{x}_{t+1} \ . \end{aligned} \tag{22}$$

By substituting (22) into (20), the terms involving the future observations $\mathbf{Y}_{t+1:T}$ can then be eliminated using (21) to yield

$$p_{t|T} = p_{t|t} \int \frac{p(\mathbf{x}_{t+1}|\mathbf{Y}_T)\, p(\mathbf{x}_{t+1}|\mathbf{x}_t)}{p(\mathbf{x}_{t+1}|\mathbf{Y}_t)}\, d\mathbf{x}_{t+1} \ , \tag{23}$$

whose denominator can be expressed in terms of the filtered distribution as

$$p(\mathbf{x}_{t+1}|\mathbf{Y}_t) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)\, p_{t|t}\, d\mathbf{x}_t \tag{24}$$

to yield the BSS formula (6).

## C. MCMC via Metropolis-Hastings

Section 4 has introduced MCMC, which uses a cleverly-chosen Markov chain to generate samples $\mathbf{z} = (\mathbf{x}_1, ..., \mathbf{x}_T)$ from the posterior $\pi(\mathbf{z}) = p(\mathbf{z} \,|\, \mathbf{Y}_T)$. The art of MCMC is in choosing

the particular ergodic Markov chain, specified by its transition probabilities $T(\mathbf{z}' \,|\, \mathbf{z})$, which has $\pi$ as a stationary distribution.

The widely used Metropolis-Hastings technique (Hastings 1970) solves this problem as follows. Suppose the chain is in state $\mathbf{z}$. Propose a new state $\mathbf{z}'$ using a density $q(\mathbf{z}' \,|\, \mathbf{z})$, and replace $\mathbf{z}$ by $\mathbf{z}'$ with probability

$$\rho(\mathbf{z} \mapsto \mathbf{z}') = \min\bigl(1, \frac{\pi(\mathbf{z}')\, q(\mathbf{z} \,|\, \mathbf{z}')}{\pi(\mathbf{z})\, q(\mathbf{z}' \,|\, \mathbf{z})}\bigr) \quad . \tag{25}$$

Then, the overall transition probability $T(\mathbf{z}' \,|\, \mathbf{z}) = q(\mathbf{z}' \,|\, \mathbf{z})\rho(\mathbf{z} \mapsto \mathbf{z}')$. Eq.(25) is the largest acceptance probability not exceeding unity in which the "detailed-balance" property holds:

$$T(\mathbf{z}' \,|\, \mathbf{z})\, \pi(\mathbf{z}) = T(\mathbf{z} \,|\, \mathbf{z}')\, \pi(\mathbf{z}') \quad .$$

That is, $\rho$ is a "valve" chosen so the amount of probability mass flowing from $\mathbf{z}'$ to $\mathbf{z}$ is in proportion $\pi(\mathbf{z})/\pi(\mathbf{z}')$ to that flowing the other way. Together with the property of recurrence, the detailed-balance condition is sufficient to ensure convergence of the sampled sum in (8) to the expectation integral.

Recurrence is a weak property: it means that every state $\mathbf{z}'$ can be reached from any state $\mathbf{z}$ eventually, with nonzero probability. When Gaussian proposals $q(\mathbf{z}' \,|\, \mathbf{z})$ are used, recurrence follows immediately. For us, $\mathbf{z}$ is multivariate, and separate Gaussian proposals can be used along various subvectors of $\mathbf{z}$ to provide recurrence. The combined chain that cyclically updates each subvector of $\mathbf{z}$ in accordance with detailed balance also preserves detailed balance.

We take a simple MCMC approach for the smoothing problem considered in this paper. Suppose the proposal strategy sweeps over $\mathbf{x}_t$ from $t = 1$ to $T$, each time proposing a new value $\mathbf{x}'_t$ in a Gaussian pattern centered about $\mathbf{x}_t$. According to the discretized dynamics of section 5, the stochastic forcing of $\mathbf{x}_t$ has a covariance of $\kappa^2\tau\mathbf{I}$, which gives an estimate of the scale for proposals. In the above language, we have chosen $q(\mathbf{z}' \,|\, \mathbf{z}) = \phi(\mathbf{x}_t - \mathbf{x}'_t, s\kappa^2\tau\mathbf{I})$, where $\phi(\mathbf{x}, \Sigma) = \exp(-\mathbf{x}^T\Sigma^{-1}\mathbf{x}/2)/(\det\Sigma)^{1/2}(2\pi)^{d/2}$, the multivariate Gaussian density in $d$ dimensions. The scale constant $s$ is chosen to give a good compromise between high acceptance probability and significant jumps. We chose $s = 1$ throughout, which gave a typical acceptance probability near 1/2. This proposal is symmetric, that is, $q(\mathbf{z}' \,|\, \mathbf{z}) = q(\mathbf{z} \,|\, \mathbf{z}')$, so this factor drops out below. In view of (25), the acceptance probability $\rho$ becomes the smaller of 1 and

$$\tilde{\rho}(\mathbf{z} \mapsto \mathbf{z}') = \frac{\pi(\mathbf{z}')\, q(\mathbf{z} \,|\, \mathbf{z}')}{\pi(\mathbf{z})\, q(\mathbf{z}' \,|\, \mathbf{z})} = \frac{p(\mathbf{x}'_t \,|\, \mathbf{x}_{t-1})\, p(\mathbf{x}_{t+1} \,|\, \mathbf{x}'_t)}{p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\, p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t)} \times \frac{p(\mathbf{y}_t \,|\, \mathbf{x}'_t)}{p(\mathbf{y}_t \,|\, \mathbf{x}_t)} \quad . \tag{26}$$

The first factor in $\tilde{\rho}$ favors a better fit of $\mathbf{x}'_t$ to its two neighbors, and the second factor rewards a better fit to the observed data. (If there was no observation at $t$, the second factor is omitted.)

Specifically, for the double-well problem, we propose to change $\mathbf{x}_t$ to $\mathbf{x}'_t$ and use (26) with $p(\mathbf{x}'_t \,|\, \mathbf{x}_{t-1}) = \phi\big(\mathbf{x}'_t - (\mathbf{x}_{t-1} + \tau g(\mathbf{x}_{t-1})), \tau\kappa^2\big)$, etc., and $p(\mathbf{y}_t \,|\, \mathbf{x}'_t) = \phi(\mathbf{y}_t - \mathbf{x}'_t, \sigma^2)$. A similar expression holds true for the L63 problem. After repeating this propose/accept process for each entry $\mathbf{x}_t$ within $\mathbf{z}$, the sweep is complete and a new draw from $\pi$ has been obtained. The collection of articles introduced by Gilks *et al.* (1996) is a good general reference on the practical aspects of MCMC.

## References


Alexander, F. J., G. L. Eyink, and J. M. Restrepo (2005). Accelerated Monte Carlo for optimal estimation of time series. *Journal of Statistical Physics*, **119**, 1331–45.

Anderson, J. L., and S. L. Anderson (1999). A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, **127**, 2741–2758.

Balgovind, R., A. Dalcher, M. Ghil, and E. Kalnay (1983). A stochastic-dynamic model for the spatial structure of forecast error statistics. *Monthly Weather Review*, **111**, 701–722.

Bengtsson, T., C. Snyder, and D. Nychka (2003). Toward a nonlinear ensemble filter for high-dimensional systems. *J. Geophys. Res.*, **108**(D24), 8775, doi:10.1029/2002JD002900.

Bennett, A. F. (1992). *Inverse Methods in Physical Oceanography.* Cambridge University Press. 346 pp.

Burgers, G., P. J. van Leeuwen, and G. Evensen (1998). Analysis scheme in the Ensemble Kalman Filter. *Monthly Weather Review*, **126**, 1719–1724.

Cohn, S. E., N. S. Sivakumaran, and R. Todling (1994). A fixed-lag Kalman smoother for retrospective data assimilation. *Monthly Weather Review*, **122**, 2838–2867.

Crisan, D. (2001). Particle filters – A theoretical perspective. In Doucet, A., N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, pages 17–41. Springer.

Daley, R. (1991). *Atmospheric Data Analysis.* Cambridge University Press. 457 pp.

Doucet, A., S. J. Godsill, and C. Andrieu (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, **10**, 197–208.

Doucet, A., N. de Freitas, and N. Gordon (2001). *Sequential Monte Carlo Methods in Practice.* Springer-Verlag.

Ehrendorfer, M., and J. J. Tribbia (1997). Optimal prediction of forecast error covariances through singular vectors. *J. Atmos. Sci.*, **54**, 286–313.

Errico, R. M. (1997). What is an adjoint model? *Bull. Am. Meteor. Soc.*, **78**, 2557–2591.

Evensen, G., and P. J. van Leeuwen (2000). An ensemble Kalman smoother for nonlinear dynamics. *Monthly Weather Review*, **128**, 1852–1867.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte-Carlo methods to forecast error statistics. *J Geophys Res*, **99**, 10143–10162.

Evensen, G. (2003). The Ensemble Kalman Filter: theoretical formulation and practical implementation. *Ocean Dyn.*, **53**, 343367.

Eyink, G. L., and J. M. Restrepo (2000). Most probable histories for nonlinear dynamics: tracking climate transitions. *Journal of Statistical Physics*, **101**, 459–472.

Gelb, A. (1974). *Applied Optimal Estimation*. MIT Press, Cambridge, MA. 374 pp.

Ghil, M. (1997). Advances in sequential estimation for atmospheric and oceanic flows. *J. Meteor. Soc. Japan*, **75**, 289–304.

Gilks, W. R., S. Richardson, and D. J. Spiegelhalter (1996). Introducing Markov chain Monte Carlo. In Gilks, W. R., S. Richardson, and D. J. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*, pages 1–19. Chapman and Hall.

Godsill, S. J., and T. Clapp (2001). Improvement strategies for Monte Carlo particle filters. In *Sequential Monte Carlo Methods in Practice*. Springer-Verlag.

Godsill, S. J., A. Doucet, and M. West (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, **99**, 156–168.

Haario, H., M. Laine, M. Lehtinen, E. Saksman, and J. Tamminen (2004). Markov chain monte carlo methods for high dimensional inversion in remote sensing. *J. Royal Stat. Soc. Ser. B*, **66**(3), 591–607.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

Houtekamer, P. L., and H. L. Mitchell (1998). Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, **126**, 796–811.

Hürzeler, M., and H. R. Künsch (1998). Monte Carlo approximations for general state space models. *Journal of Computational and Graphical Statistics*, **7**, 175–193.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press, New York.

Kalnay, E. (2003). *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, Cambridge, UK. 341 pp.

Keppenne, C. L., and M. M. Rienecker (2002). Initial testing of a massively parallel ensemble Kalman filter with the Poseidon isopycnal ocean general circulation model. *Monthly Weather Review*, **130**, 2951–2965.

Kim, S., G. L. Eyink, J. M. Restrepo, F. J. Alexander, and G. Johnson (2003). Ensemble filtering for nonlinear dynamics. *Monthly Weather Review*, **131**, 2586–2594.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, **5**, 1–25.

Lawson, W. G., and J. A. Hansen (2004). Implications of stochastic and deterministic filters as ensemble-based data assimilation methods in varying regimes of error growth. *Monthly Weather Review*, **132**, 1966–1981.

Liu, J. S., and R. Chen (1998). Sequential Monte Carlo methods for dynamic systems. *Jour. Amer. Statist. Assoc.*, **93**(443), 1032–1044.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *J. Atmos. Sci.*, **20**, 130–141.

McLachlan, G., and D. Peel (2000). *Finite Mixture Models*. Wiley.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, **21**, 1087.

Miller, R. N., and L. L. Ehret (2002). Ensemble generation for models of multimodal systems. *Monthly Weather Review*, **130**, 2313–2333.

Miller, R. N., M. Ghil, and F. Gauthiez (1994). Advanced data assimilation in strongly nonlinear dynamical systems. *J Atmos Sci*, **51**, 1037–1056.

Miller, R. N., E. F. Carter, and S. T. Blue (1999). Data assimilation into nonlinear stochastic models. *Tellus*, **51A**, 167–194.

Molteni, F., R. Buizza, T. N. Palmer, and T. Petroliagis (1996). The ECMWF ensemble prediction system: Methodology and validation. *Quart. J. Roy. Meteor. Soc.*, **122**, 73–119.

Ott, E., B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. J. Patil, and J. A. Yorke (2004). A local ensemble Kalman filter for atmospheric data assimilation. *Tellus*, **56A**, 415–428.

Pham, D. T. (2001). Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly Weather Review*, **129**, 1194–1207.

Robert, C. P., and G. Casella (2005). *Monte Carlo Statistical Methods*. Springer.

Tierney, L. (1996). Introduction to general state-space Markov chain theory. In Gilks, W. R., S. Richardson, and D. J. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*, pages 59–74. Chapman and Hall.

Toth, Z., and E. Kalnay (1993). Ensemble forecasting at NMC: The generation of perturbations. *Bull. Amer. Meteor. Soc.*, **74**, 2317–2330.

Turmon, M., J. Pap, and S. Mukhtar (2002). Statistical pattern recognition for labeling solar active regions: Application to SoHO/MDI imagery. *Astrophysical Journal*, **568**(1), 396–407.

van Leeuwen, P. J., and G. Evensen (1996). Data assimilation and inverse methods in terms of a probabilistic formulation. *Monthly Weather Review*, **124**, 2898–2913.

van Leeuwen, P. J. (2003). A variance-minimizing filter for large-scale applications. *Monthly Weather Review*, **131**, 2071–2084.

Wunsch, C. (1996). *The Ocean Circulation Inverse Problem*. Cambridge University Press. 437 pp.

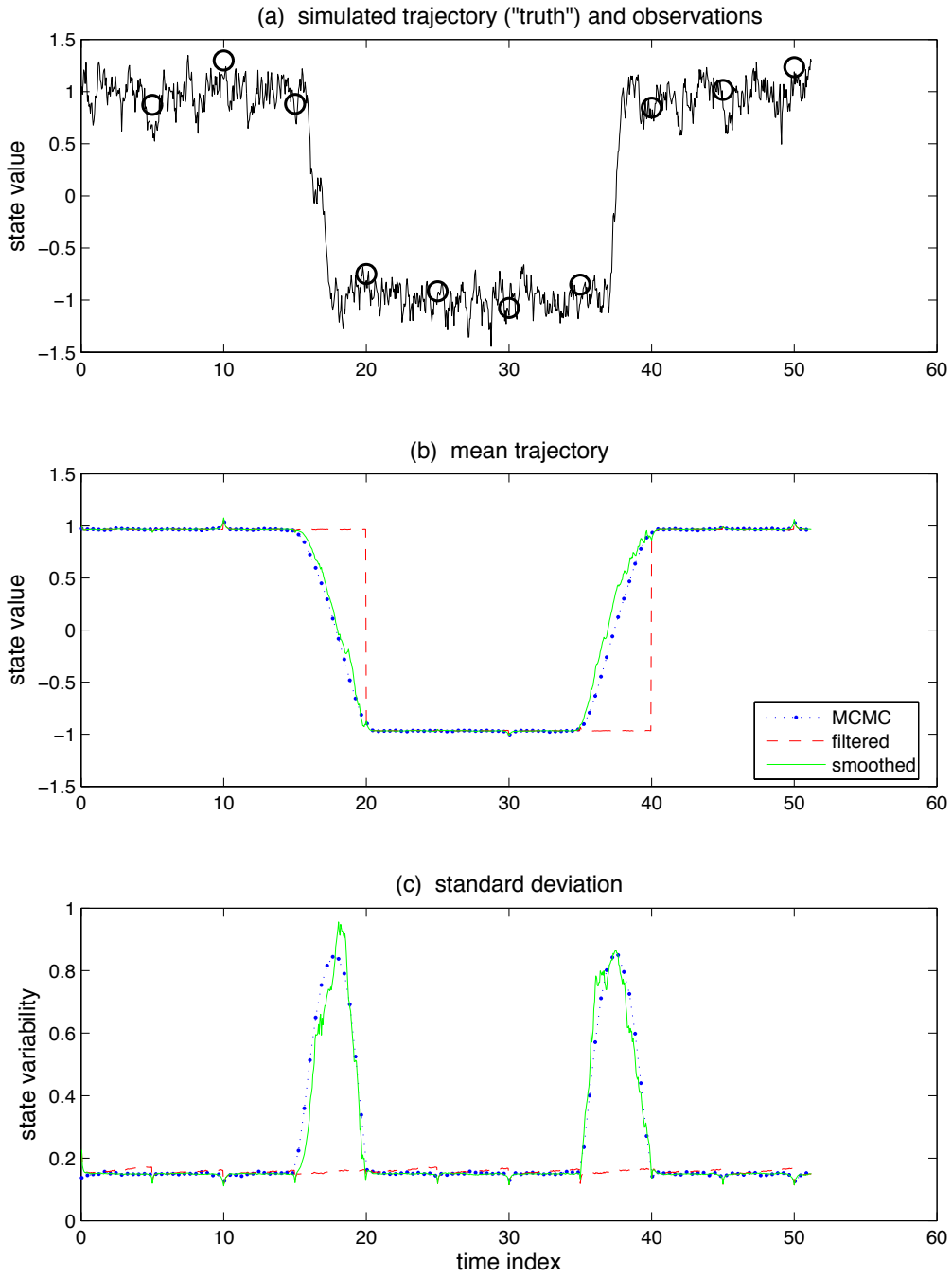


**Figure 1**. Algorithm performance for the double-well case. (a) Sample trajectory used as the truth (solid line) with sparse and noisy observations (open circles). (b) Mean trajectories from MCMC (2000 samples after $10^5$ iterations, dots and filled circles), resampled particle filter (RPF with $10^4$ particles, dashed line), and particle smoother (RPF followed by BSS with $10^4$ particles, solid line). (c) Standard deviations of the corresponding algorithms.

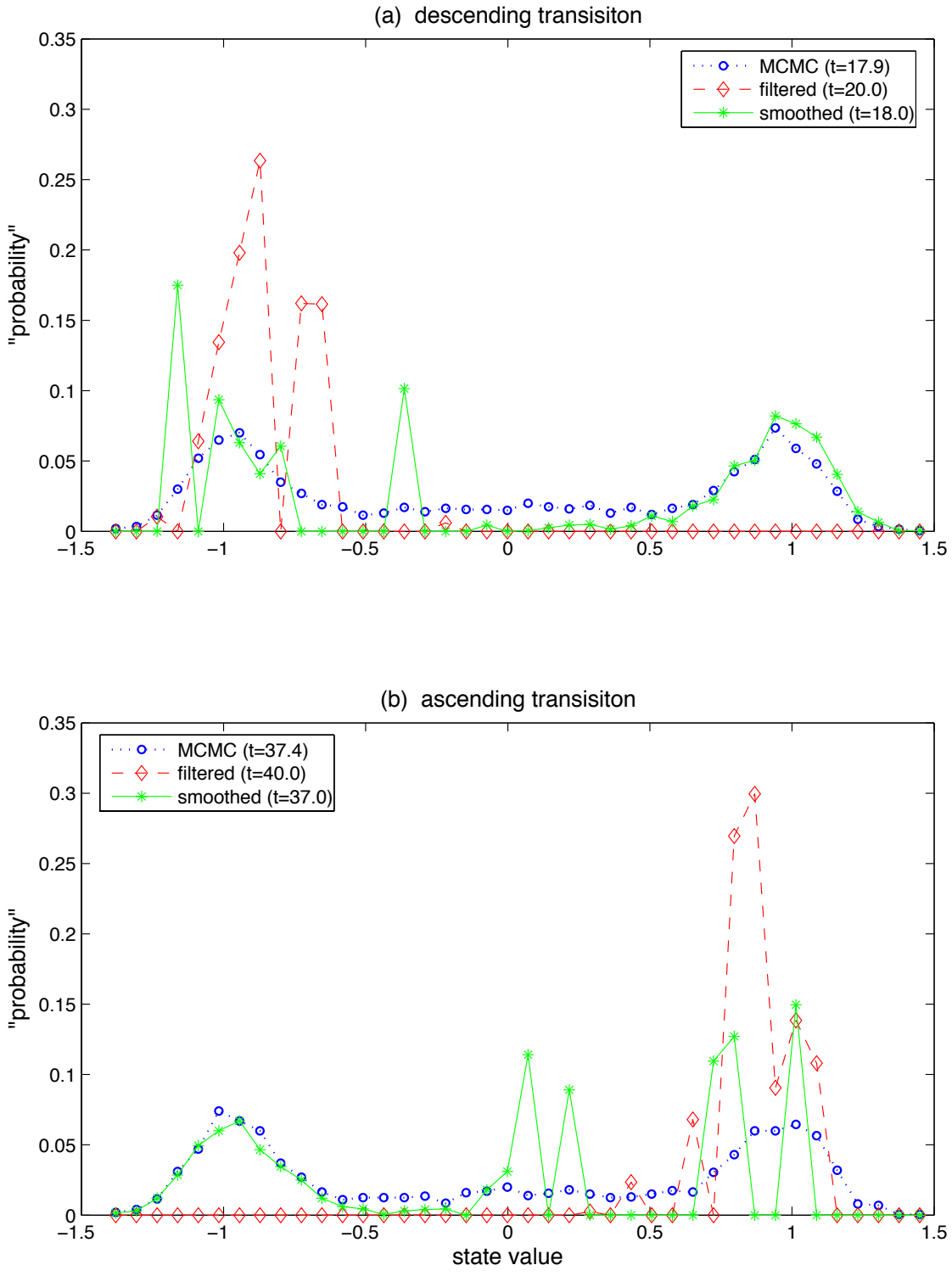


**Figure 2**. Normalized histograms of the state values at the (a) descending and (b) ascending zero-crossings in Fig. 1b. The line styles are the same as in Figs. 1b and c.

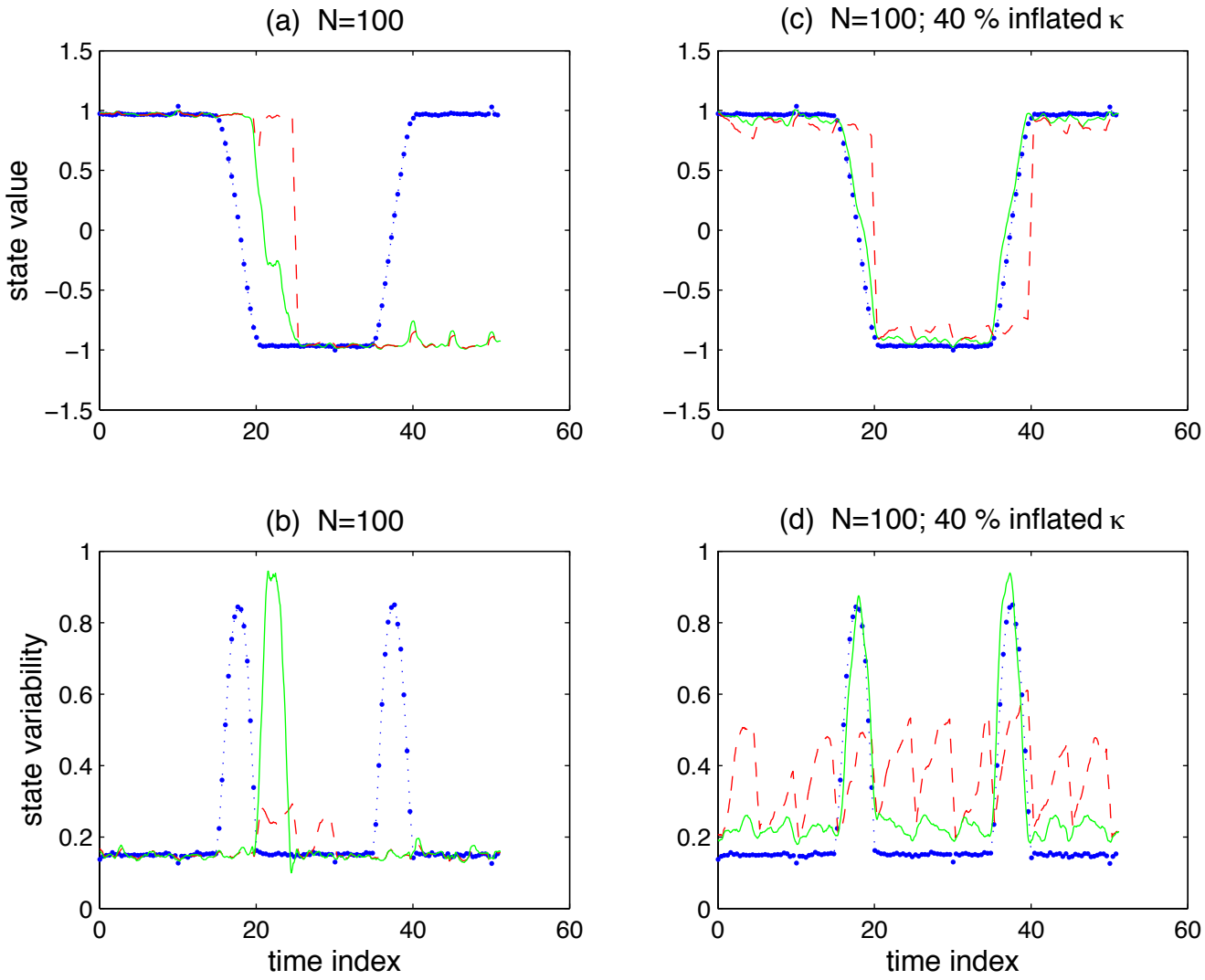


**Figure 3**. Algorithm performance for a smaller numbers of particles. (a, b) Similar to Figs. 1b and c, respectively, except that the particle number $N$ is only 100. (c, d) Similar to (a) and (b), respectively, except that the system noise $\kappa$ is artificially inflated by 40%. The line styles are the same as in Figs. 1b and c.

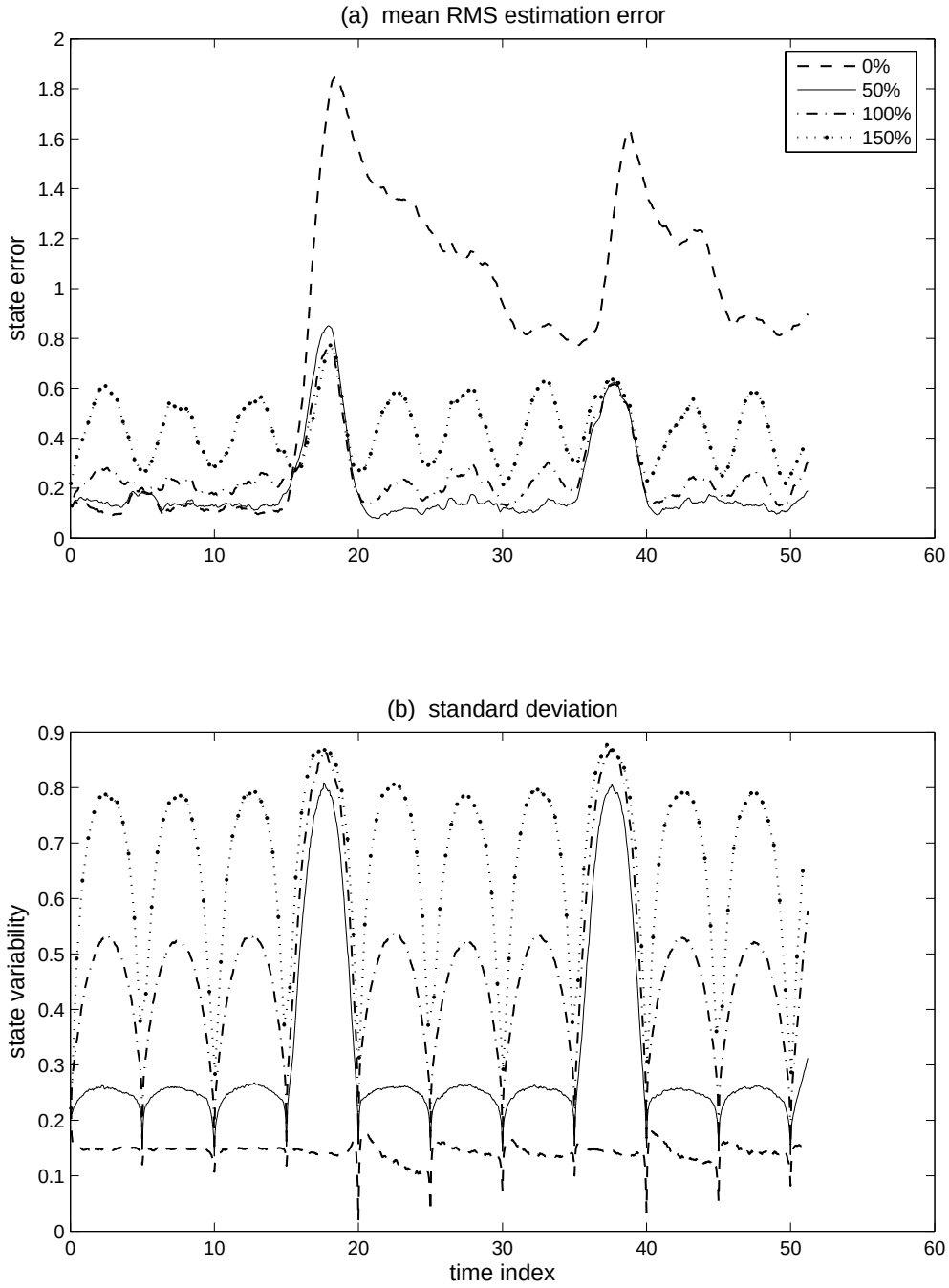


**Figure 4**. Effect of noise-amplification factor on algorithm performance. (a) Mean estimation errors in the double-well problem of Fig. 1. The RPF+BSS smoother is used with the indicated level of increase in the system noise variance value $\kappa$; note that no increase in $\kappa$ (0%) leads to a high level of estimation error. (b) Ensemble variability similar to Fig. 1c but corresponding to the four smoothers in panel (a) above; note that the smoother with a 50% increase in $\kappa$ shows the best match between the formal and empirical errors.

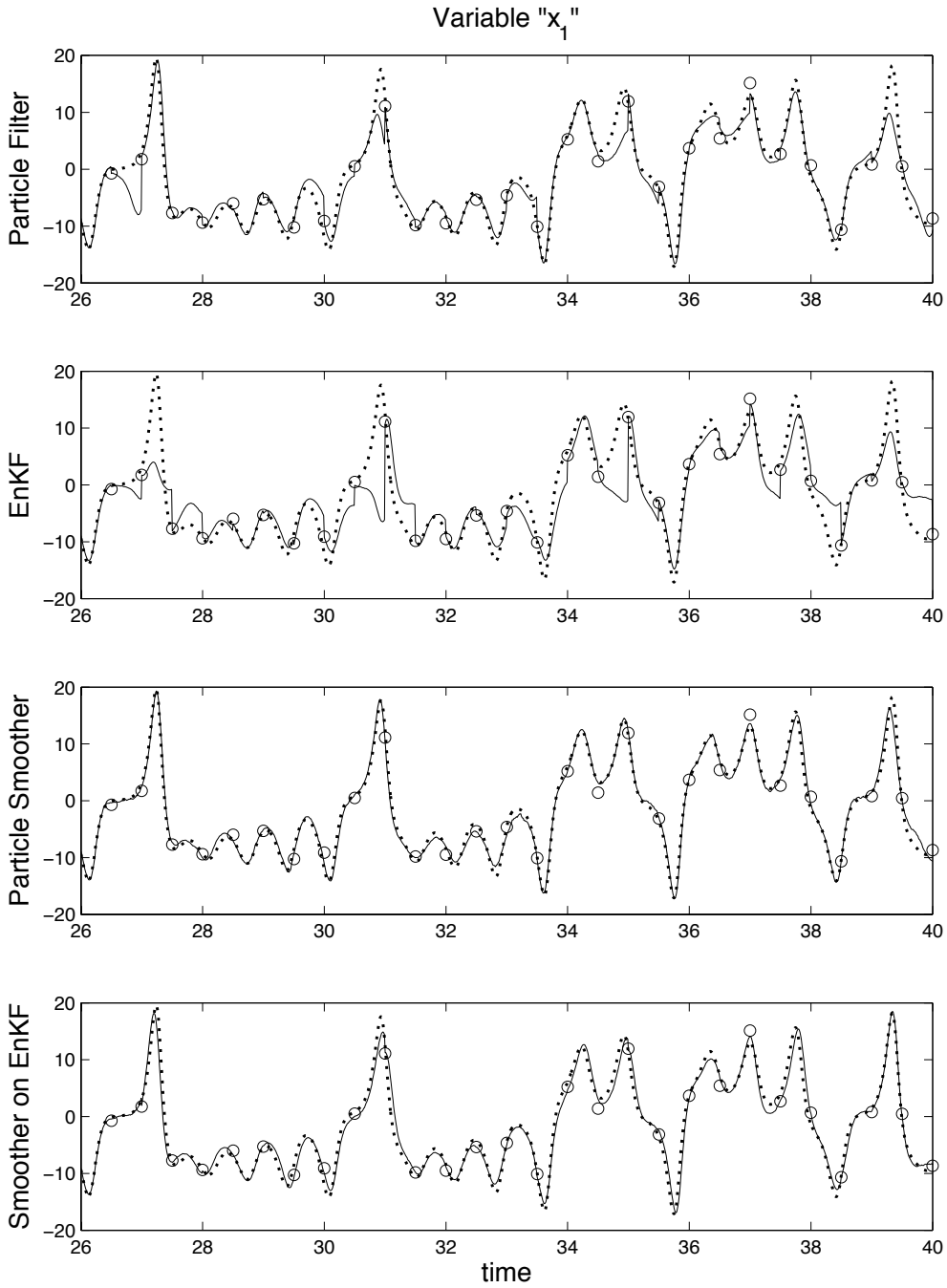


**Figure 5**. Algorithm performance for the Lorenz model (L63). The $x_1$ variable of a trajectory (dotted line, each panel) and its noisy observations (open circles). Estimates (solid lines) of the trajectory by the Resampled Particle Filter (RPF), Ensemble Kalman Filter (EnKF), Resampled Particle Smoother, and the Smoother applied to the EnKF result are shown in order from top to bottom panel.

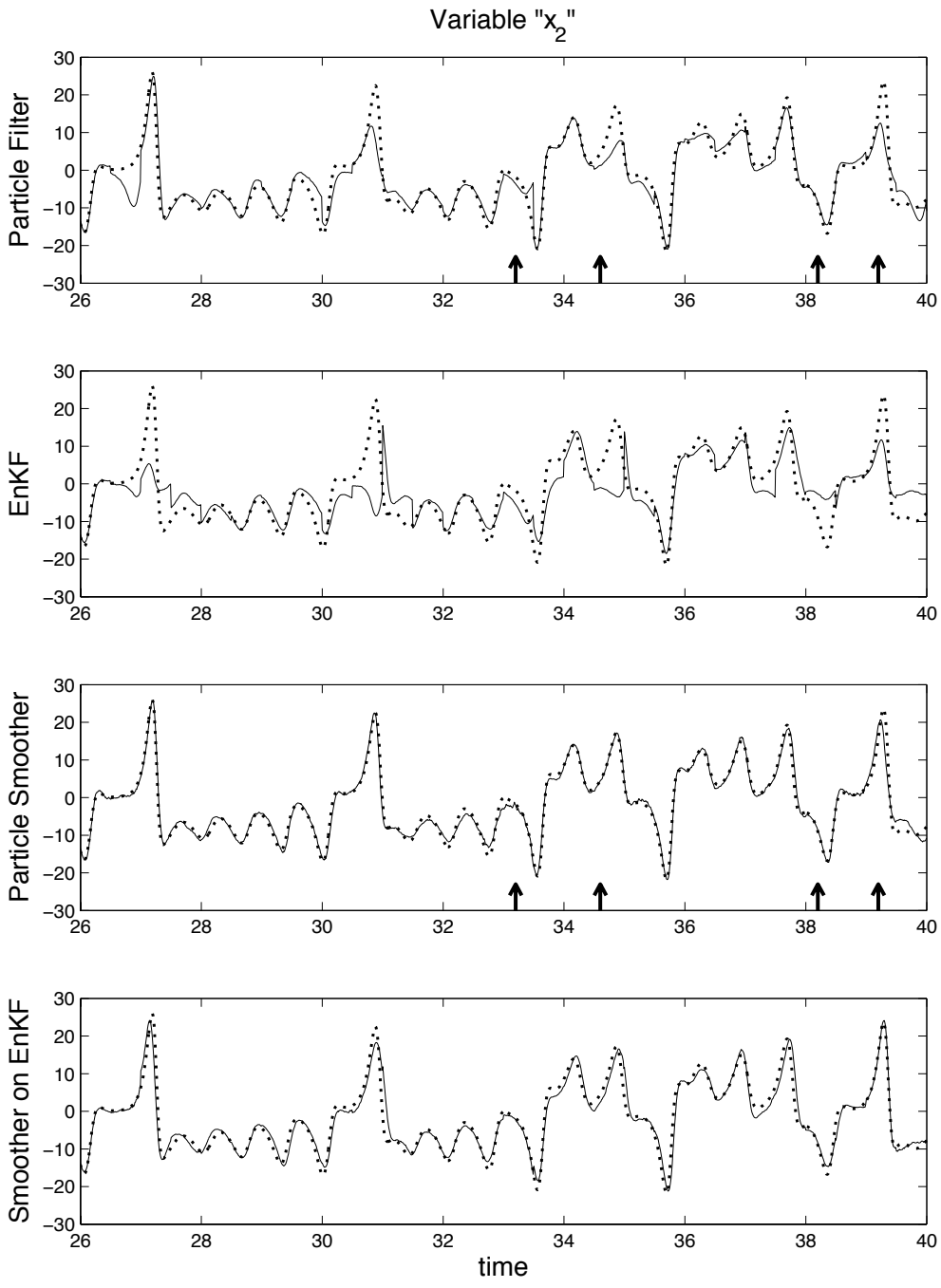


**Figure 6**. The same as Fig. 5 except that the $x_2$ variable is shown; note that the $x_2$ variable is not directly observed (see text). The arrows indicate the epochs of the order statistics shown in Fig. 9.

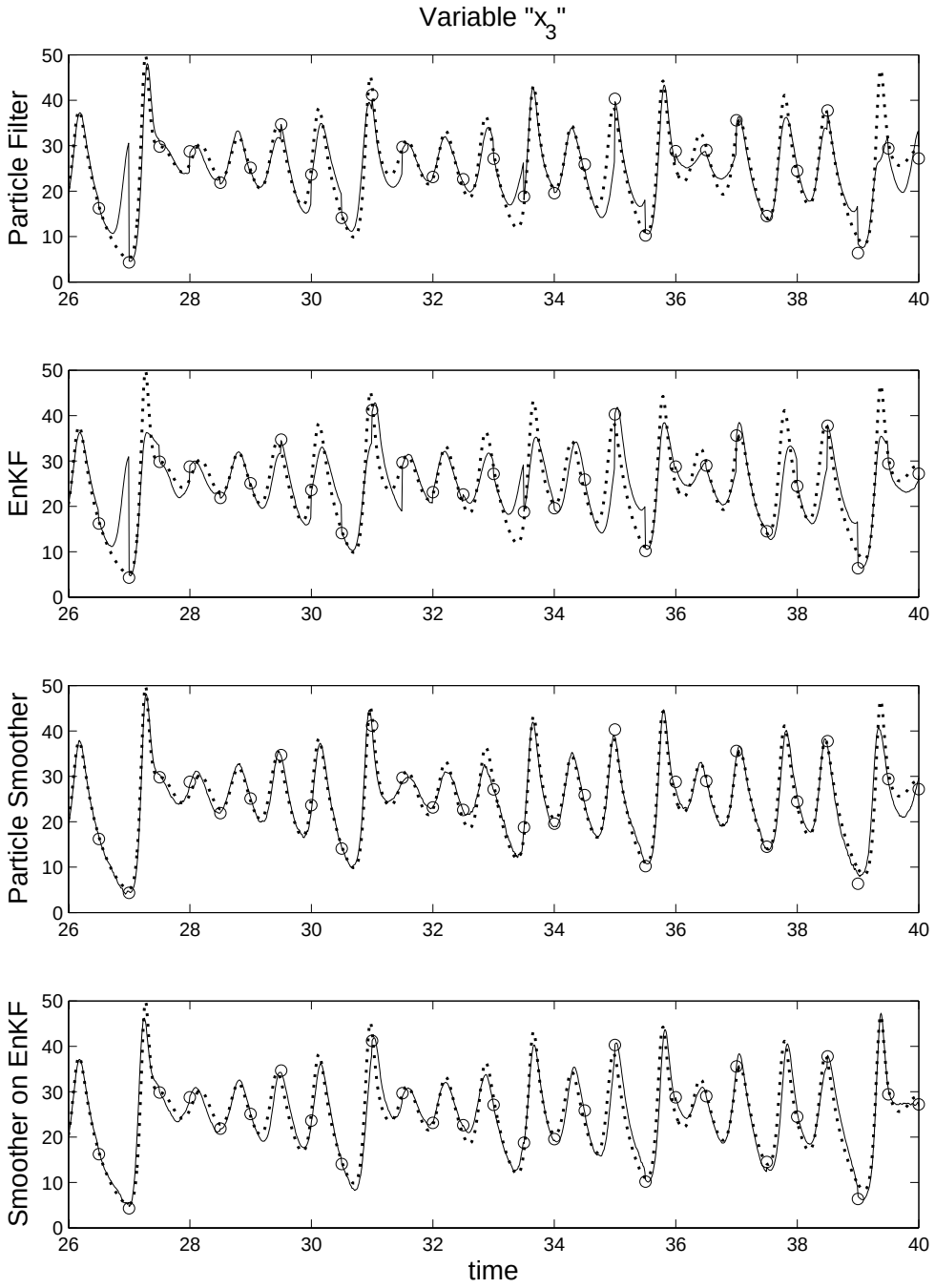


**Figure 7**. The same as Fig. 5 except that the $x_3$ variable is shown.

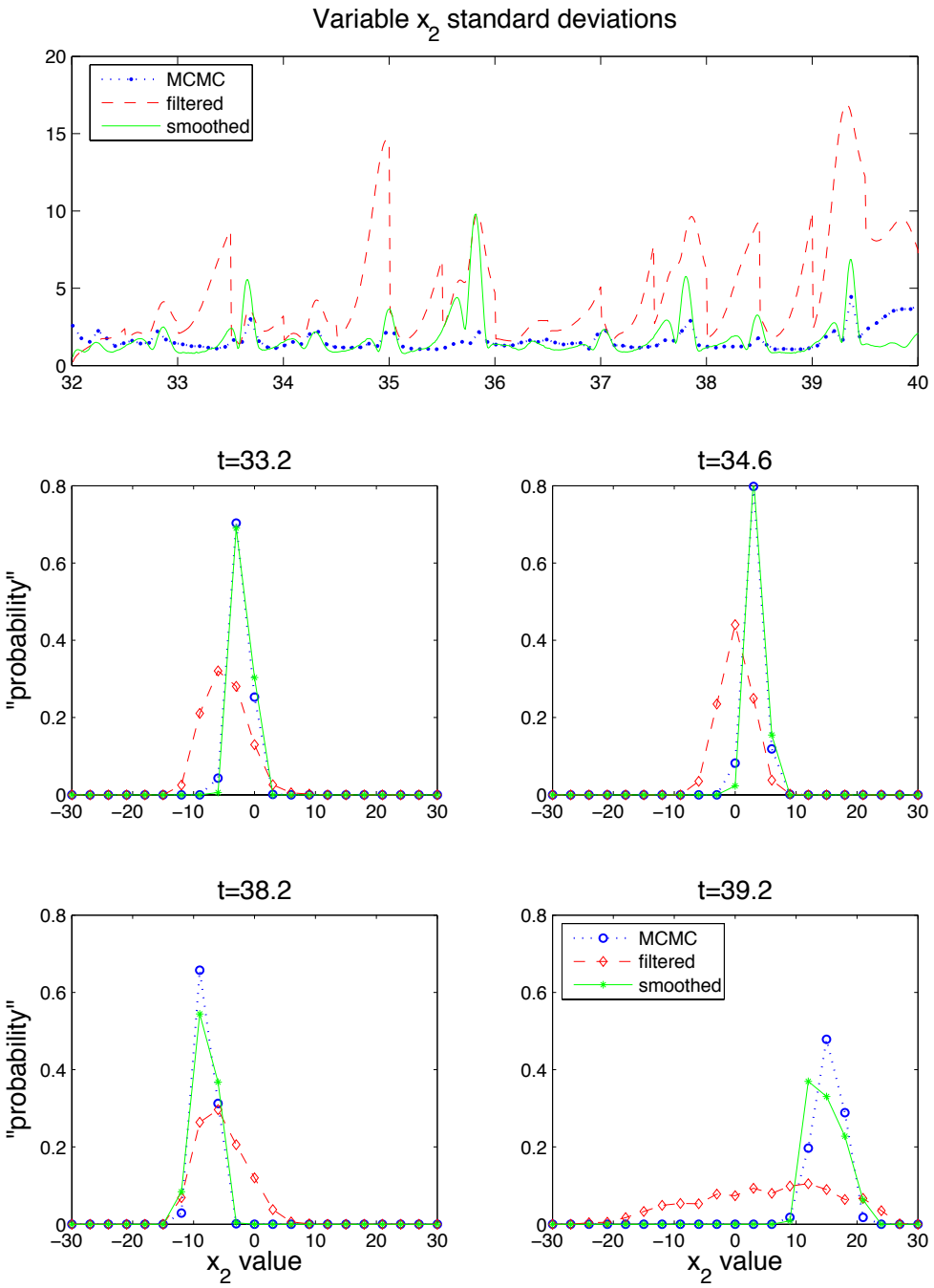


**Figure 8**. The ensemble standard deviations for the MCMC, RPF (filtered), and RPF-BSS (smoothed) methods, applied to the L63 problem (top panel). The corresponding posterior distribution, estimated as a normalized histogram, of the $x_2$ variables at selected epochs (arrows in Fig. 6). The line styles are the same as in Figs. 1b and c.

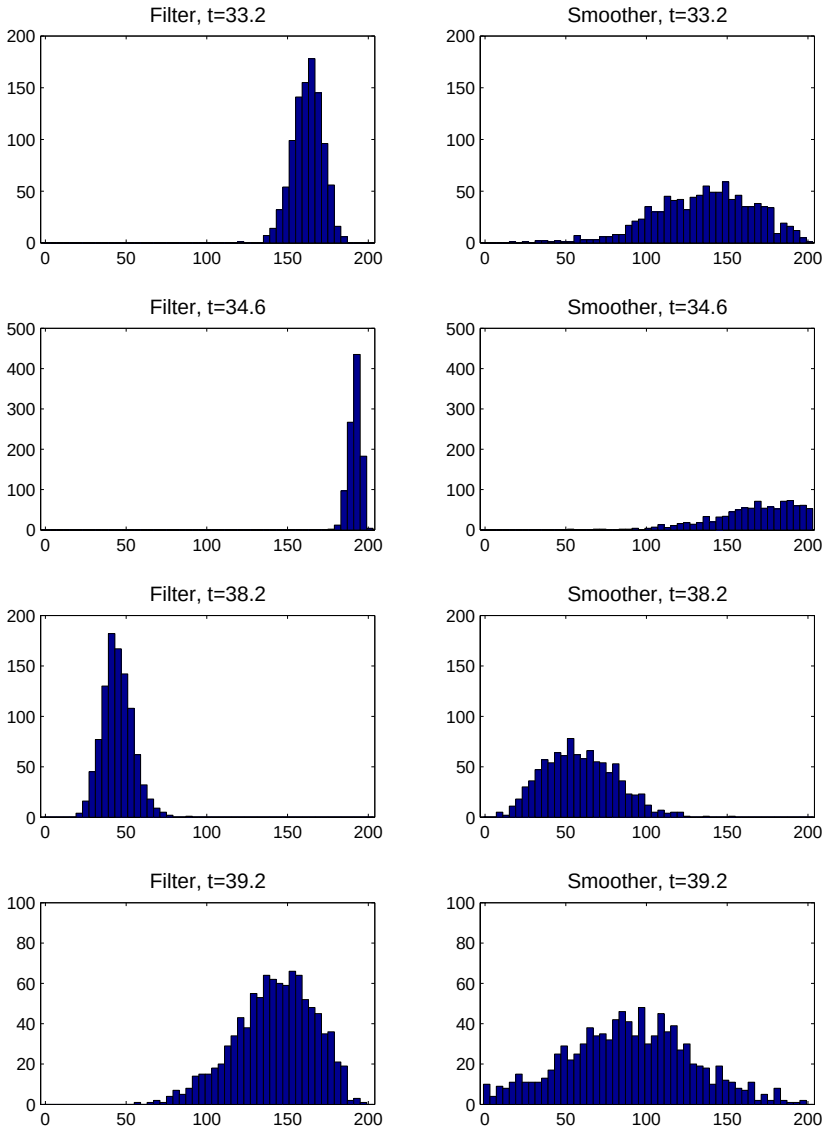


**Figure 9**. Order statistic histograms for the value of the $x_2$ variable in the L63 case at selected epochs (arrows in Fig. 6) in the Resampled Particle Filter and Smoother ensembles. The ensemble size was 200, and 1000 simulations were used to compile the histogram. Note that unbiased ensembles would tend to yield a uniform distribution for the order statistic (i.e., a flat histogram). The ensemble under/*over*-estimates the true value as the histogram is biased toward the right/*left* (higher/*lower* bin number). As expected, the smoother gives a more uniform distribution than the filter.

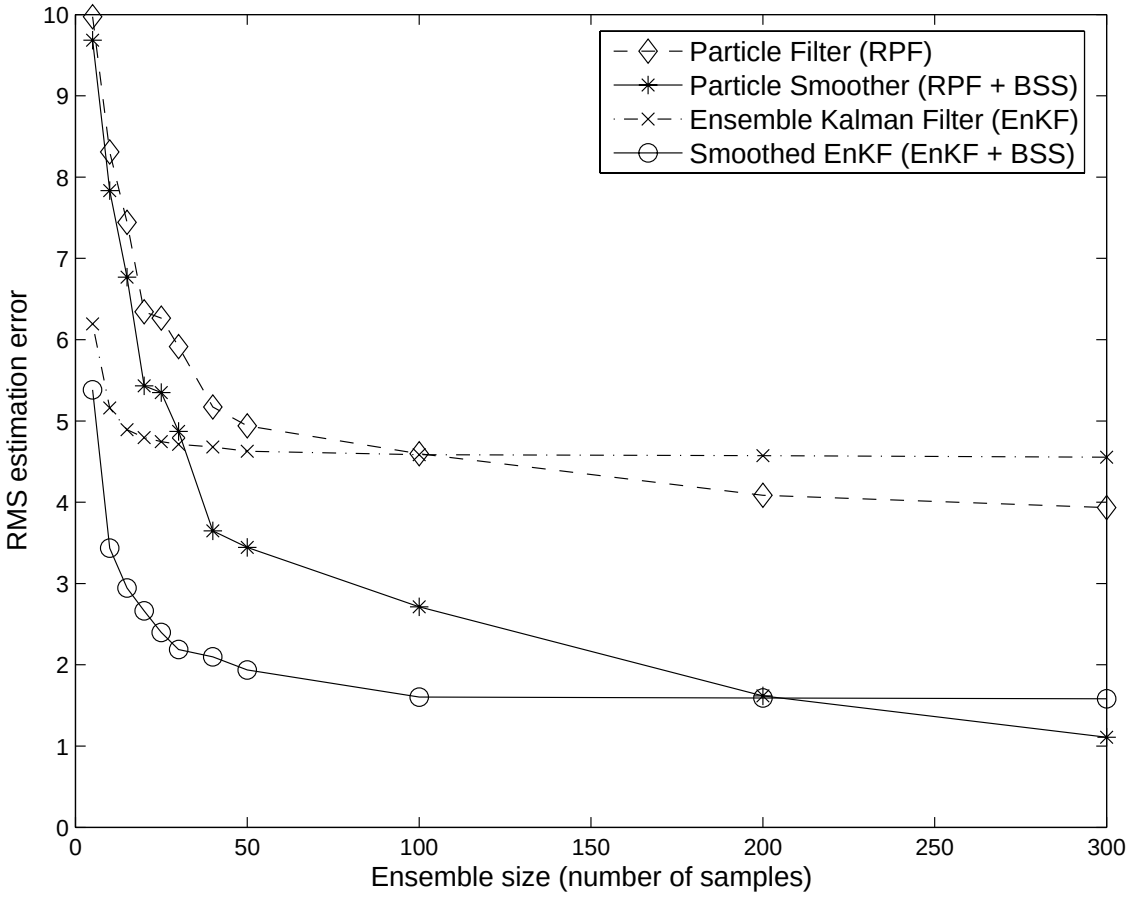


**Figure 10**. Root-mean-square (RMS) estimation errors for the L63 problem, as a function of the ensemble size. Each of the four estimation schemes indicated is executed repeatedly for 50 times, with different observation noise realizations, to obtain the mean error displayed. When the ensemble size is extremely small, the parametric approach (EnKF) performs better than the non-parametric approaches (Particle Filter and Smoother). The accuracy of the parametric EnKF approach does not improve significantly when the ensemble size is increased over 30, indicating that its assumption of Gaussian distribution is not adequate.